In [4]:

```
import pandas as pd

path = r'D:\Projects\Artificial Intelligence\IMDB Dataset.csv'
df = pd.read_csv(path)

df.head()
```

Out[4]:

	review	sentiment
0	One of the other reviewers has mentioned that ...	positive
1	A wonderful little production. <br /><br />The...	positive
2	I thought this was a wonderful way to spend ti...	positive
3	Basically there's a family where a little boy ...	negative
4	Petter Mattei's "Love in the Time of Money" is...	positive

In [5]:

```
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer

vect = CountVectorizer()
docs = np.array(['This is first project i.e Sentiment Analysis'])

bag = vect.fit_transform(docs)
```

In [6]:

```
print(vect.vocabulary_)
```

```
{'this': 5, 'is': 2, 'first': 1, 'project': 3, 'sentiment': 4, 'analysis': 0}
```

In [7]:

```
print(bag.toarray())
```

```
[[1 1 1 1 1 1]]
```

In [8]:

```
from sklearn.feature_extraction.text import TfidfTransformer
np.set_printoptions(precision = 2)
tfidf = TfidfTransformer(use_idf = True, norm='l2', smooth_idf = True)
print(tfidf.fit_transform(bag).toarray())
```

```
[[0.41 0.41 0.41 0.41 0.41 0.41]]
```

In [9]:

```
import nltk
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\pc2\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Out[9]:

```
True
```

In [10]:

```
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf = TfidfVectorizer(
    use_idf = True,
    norm = 'l2',
    smooth_idf = True)
y = df.sentiment.values
X = tfidf.fit_transform(df['review'].values.astype('U'))
```

In [11]:

```
from sklearn.model_selection import train_test_split
```

In [12]:

```
X_train,X_test,y_train,y_test = train_test_split(X,y,random_state=1,test_size = 0.5, shuffle=False)
```

In [13]:

```
import pickle
from sklearn.linear_model import LogisticRegressionCV
clf = LogisticRegressionCV(cv=5,
                           scoring= 'accuracy',
                           random_state = 0,
                           n_jobs = -1,
                           verbose = 3,
                           max_iter = 300).fit(X_train,y_train)

saved_model = open('saved_model.sav', 'wb')
pickle.dump(clf, saved_model)
saved_model.close()
```

```
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 4 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:  7.0min remaining: 10.6min
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:  8.9min finished
```

In [17]:

```
filename = 'saved_model.sav'
saved_clf = pickle.load(open(filename, 'rb'))
```

```
saved_clf.score(X_test,y_test)
```

Out[17]:

```
0.89712
```

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js